Computer Organization and Structure

Homework #5
Due: 2014/12/30

1. For a direct-mapped cache design with 32-bit address, the following bits of the address are used to access the cache.

	Tag	**Index**	**Offset**
a.	31-10	9-4	3-0
b.	31-13	12-5	4-0

   a. What is the cache line size (in words)?
   b. How many entries does the cache have?
   c. What is the ratio between total bits required for such a cache implementation over the data storage bits?

   Starting from power on, the following byte-addressed cache references are recorded.

Address	0	4	16	132	232	160	1024	30	140	3100	180	2180

   a. How many blocks are replaced?
   b. What is the hit ratio?
   c. List the final state of the cache, with each valid entry represented as a record of <index, tag, data>.

2. Here is a series of address references given as word addresses: 17, 9, 6, 43, 17, 56, 5, 10, 6, 14, 7. Show the hits and misses and final cache contents for the following different types of caches, respectively. Assume LRU (Least Recently Used) replacement.

   a. A direct-mapped cache with eight one-word blocks.
   b. A two-way set-associative cache with eight one-word blocks.
   c. A four-way set-associative cache with eight one-word blocks.
   d. A fully associative cache with eight one-word blocks.
   e. A fully associative cache with two four-word blocks.

3. In general, cache access time is proportional to capacity. Assume that main memory accesses take 70ns and that memory accesses are 36% of all instructions. The following table shows data for L1 caches attached to each of two processors P1 and P2.

		L1 size	L1 miss rate	L1 hit time
a.	P1	1KB	11.4%	0.62ns
	P2	2KB	8.0%	0.66ns
b.	P1	8KB	4.3%	0.96ns
	P2	16KB	3.4%	1.08ns

   a. Assuming that the L1 hit time determines the cycle times for P1 and P2, what are

their respective clock rates?

b. What is the AMAT for each of P1 and P2?
c. Assuming a base CPI of 1.0, what is the total CPI for each of P1 and P2? Which processor is faster?

For the next three sub-problems, we will consider the addition of an L2 cache to P1 to presumably make up for its limited L1 cache capacity. Use the L1 cache capacities and hit times from the previous table when solving these sub-problems. The L2 miss rate indicate is its local miss rate.

	L2 size	L2 miss rate	L2 hit time
a.	512KB	98%	3.22ns
b.	4MB	73%	11.48ns

d. What is the AMAT for P1 with the addition of an L2 cache? Is the AMAT better or worse with the L2 cache?
e. Assuming a base CPI of 1.0, what is the total CPI for P1 with the addition of an L2 cache?
f. Which processor is faster, now that P1 has an L2 cache? If P1 is faster, what miss rate would P2 need in its L1 cache to match P1's performance? If P2 is faster, what miss rate would P1 need in its L1 cache to match P2's performance?

4. There are many different design parameters that are important to a cache's overall performance. The table below lists parameters for different direct-mapped cache designs.

	Cache data size	**Cache block size**	**Cache access time**
a.	64 KB	1 word	1 cycle
b.	64 KB	2 word	2 cycle

a. Calculate the total number of bits required for the cache listed in the table, assuming a 32-bit address. Given that total size, find the total size of the closest direct-mapped cache with 16-word blocks of equal size or grater. Explain why the second cache, despite its larger data size, might provide slower performance than the first cache.
b. Generate a series of read requests that have a lower miss rate on a 2 KB two-way set associative cache than the cache listed in the table. Identify one possible solution that would make the cache listed in the table have an equal or lower miss rate than the 2 KB cache. Discuss the advantages and disadvantages of such a solution.
c. (Block address) modulo (Number of blocks in the cache) shows the typical method to index a direct-mapped cache. Assuming a 32-bit address and 1024 blocks in the cache, consider a different indexing function, specially (Block address[31:27] XOR Block address[26:22]). Is it possible to use this to index a direct-mapped cache? If so, explain why and discuss any changes that might need to be made to the cache. If it is not possible, explain why.

5. To support multiple virtual machines, two levels of memory virtualization are needed. Each virtual machine still controls the mapping of virtual address (VA) to physical address (PA), while the hypervisor maps the physical address (PA) of each virtual

machine to the actual machine address (MA). To accelerate such mappings, a software approach called "shadow paging" duplicates each virtual machine's page tables in the hypervisor, and intercepts VA to PA mapping changes to keep both copies consistent. To remove the complexity of shadow page tables, a hardware approach called nested page table (or extended page table) explicitly support two classes of page tables (VA⇒PA and PA⇒MA) and can walk such tables purely in hardware. Consider the following sequence of operations:

(1) Create process; (2) TLB miss; (3) page fault; (4) context switch;

a. What would happen for the given operation sequence, for shadow page table, and nested page table respectively?
b. Assuming an x86-based four-level page table in both guest and nested page table, how many memory references are needed to service a TLB miss ofr native versus nested page table?
c. Among TLB miss rate, TLB miss latency, page fault rate, and page fault handler latency, which metrics are more important for shadow page table? What are important for nested page table?

The following table shows parameters for a shadow paging system.

TLB misses per 1000 instruction	**NPT TLB miss latency**	**Page faults per 1000 instruction**	**Shadowing page fault overhead**
0.2	200 cycles	0.001	30000 cycles

d. For a benchmark with native execution CPI of 1, what are the CPI numbers if using shadow page tables versus NPT (assuming only page table virtualization overhead)?
e. What techniques can be used to reduce page table shadowing induced overhead?
f. What techniques can be used to reduce NPT induced overhead?